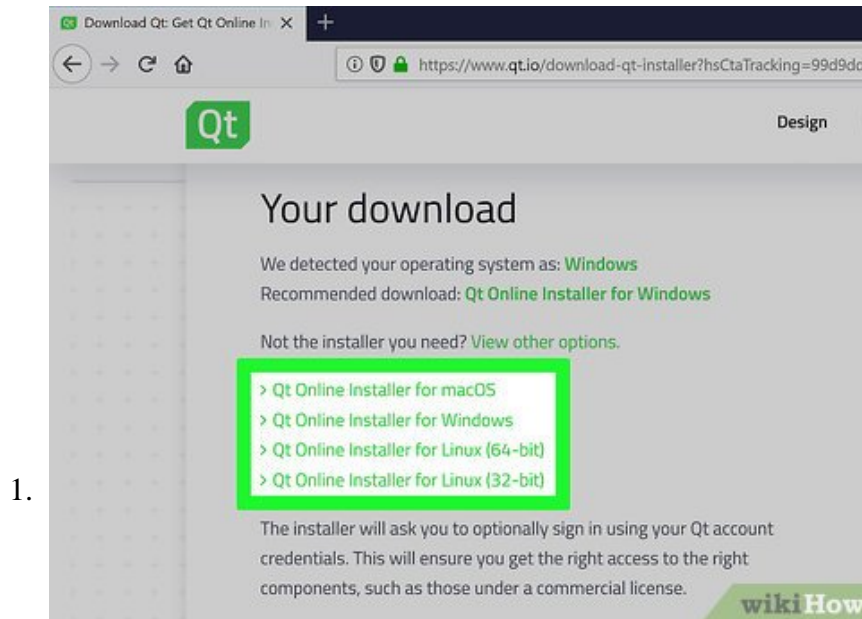


How to Install Qt SDK on Microsoft Windows

The Qt Software Development Kit (SDK) is a cross-platform application framework that is widely used for developing application software with a graphical user interface (GUI). It is a portable cross platform application user interface...

Method 1 of 6:

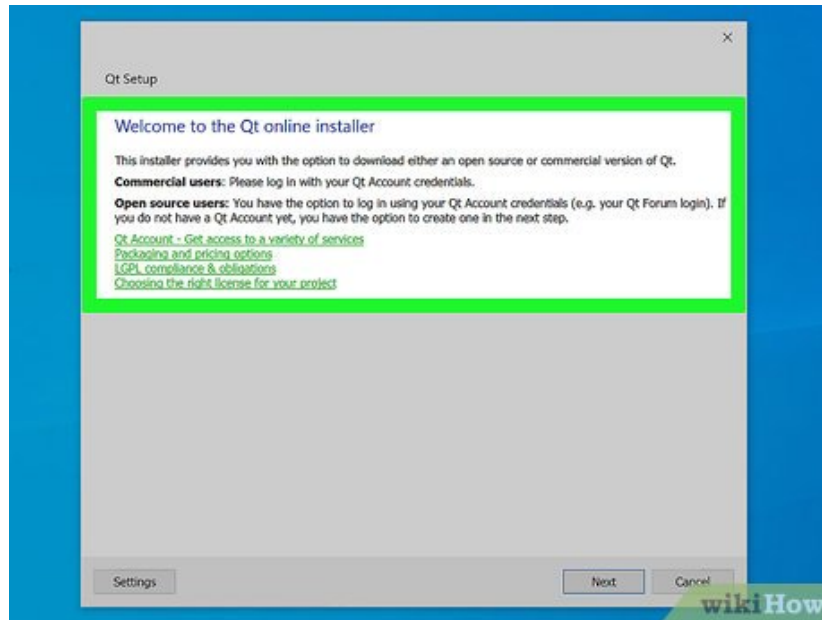
Qt 4.8 SDK Installation Instructions:



In order to setup the development environment for the Qt SDK we will have to obtain the Qt SDK. Preparing your Windows operating system for Qt application development

1. Download the Qt SDK. Select the Windows version and be prepared for a long download depending on your Internet speed. I would suggest you select the off-line installation of the Qt SDK if you do not have a very fast internet connection. The complete Qt SDK for the Windows operating system is 1.7 GB which can take over 6 hours on a slow connection.

2.

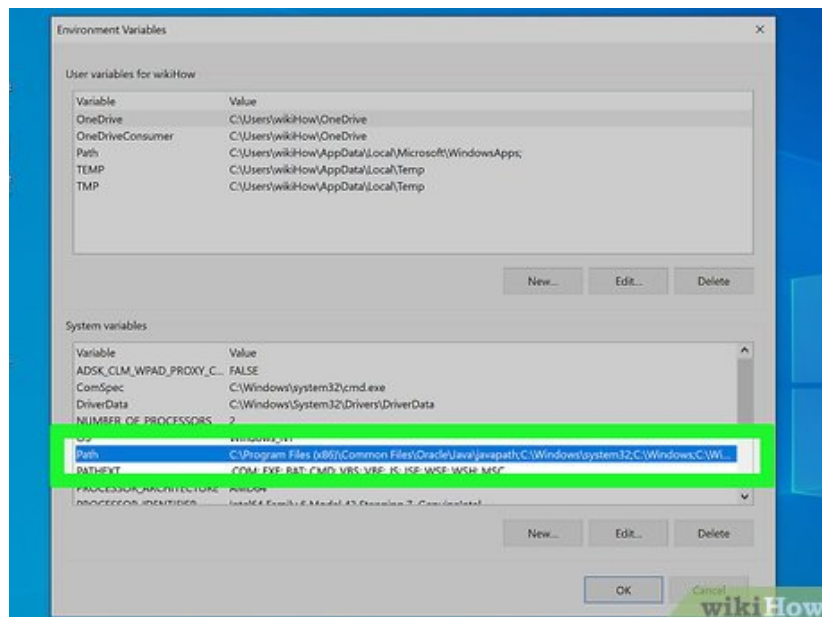


Install the Qt SDK by clicking on the Qt SDK executable. Once the Qt SDK executable is installed you will need to edit the Windows system PATH so that the Windows operating system can find the Qt commands from the command line. Be careful editing your Windows system PATH.

Method 2 of 6:

Edit the Windows Vista/Windows 7 PATH:

1.



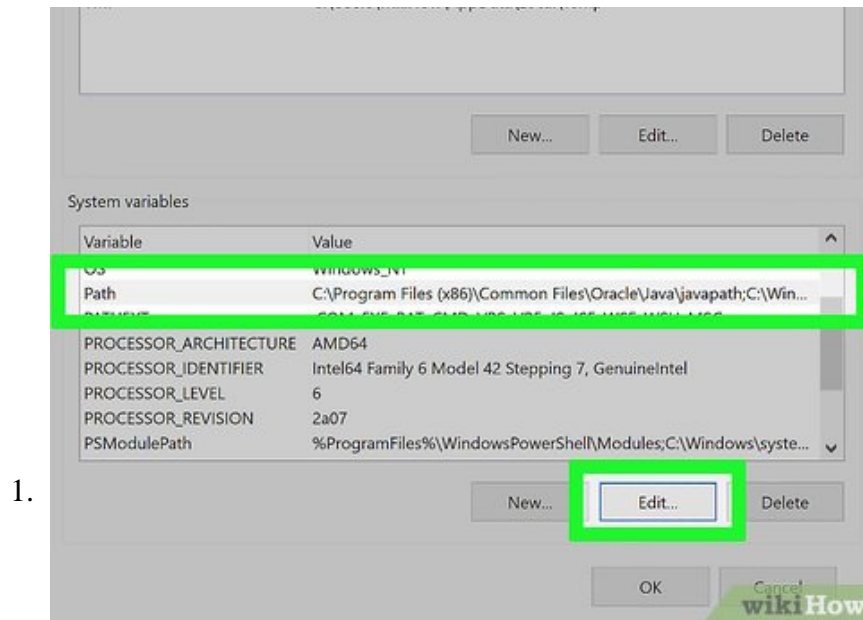
Follow the steps below.

1. Click on the Start Button
2. Click on Control Panel
3. Click on System and Maintenance

4. Click on System
5. Click on Advanced System Setting
6. Click on Environmental Variables
7. Click on PATH to edit system wide variables
8. Click OK when finished

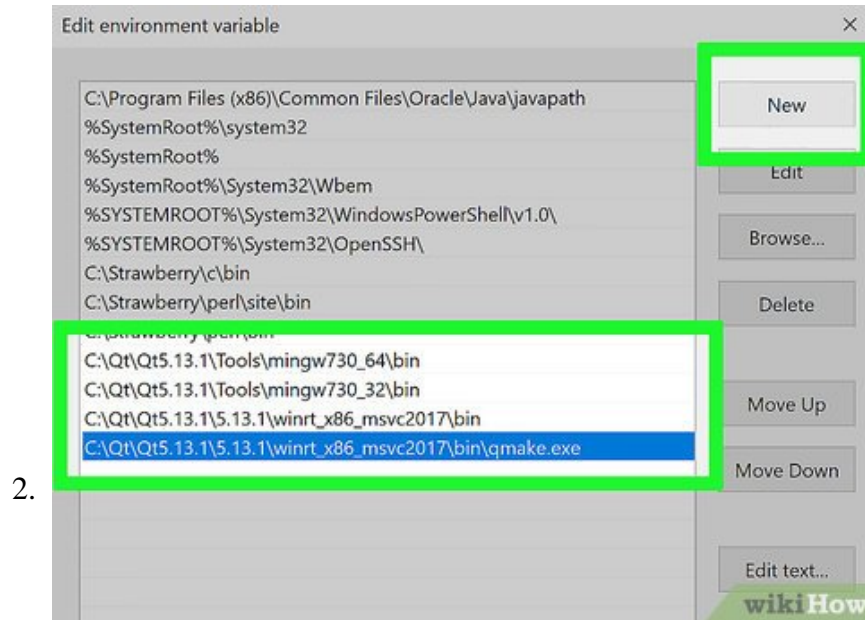
Method 3 of 6:

Edit the Windows 8 PATH:



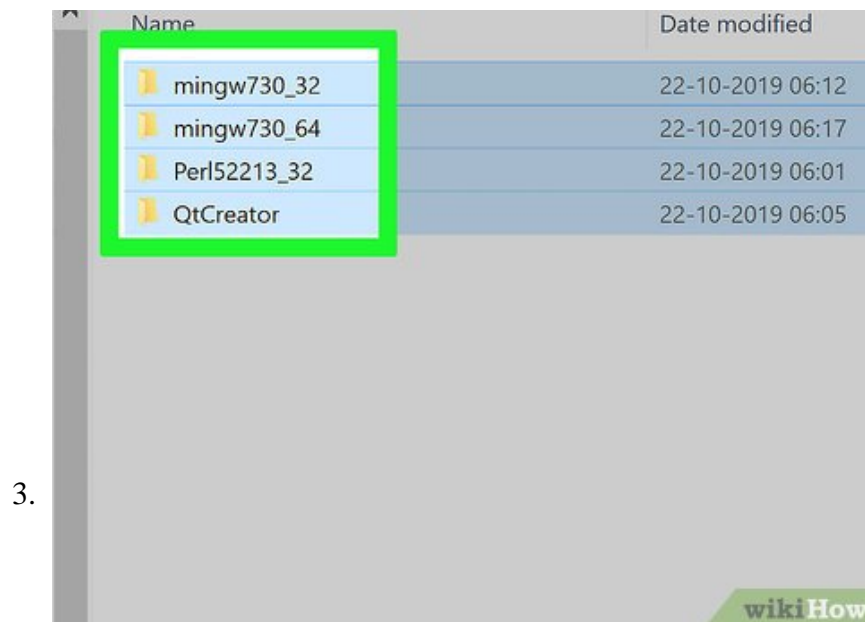
Follow the steps below.

1. Click on the Folder icon located on the bottom task bar next to the Internet Explorer icon
2. Scroll down to Computer
3. Right click with your mouse on Properties
4. Click on Advanced System Setting
5. Click on Environmental Variables
6. Click on PATH to edit system wide variables
7. Click OK when finished



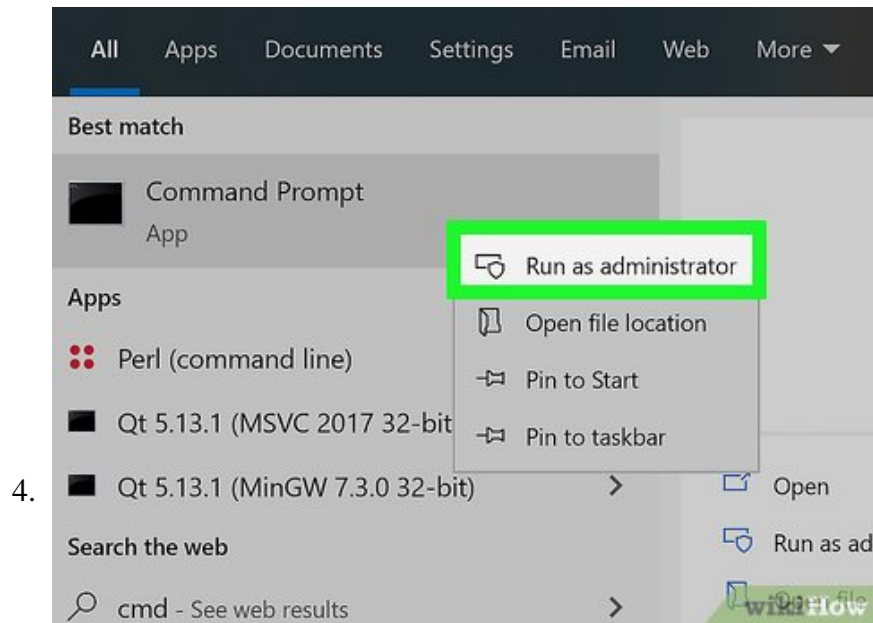
Add the following to the Windows system PATH

1. **Type/Copy/Paste:** ;C:\QtSDKmingwbin;C:\QtSDKDesktopQt4.8.1mingwbin;
2. this will set and notify the Windows operating system to compile Qt SDK applications from the command line. The numbers **4.8.1** denote the version number of the Qt SDK which changes with each new upgrade of the Qt SDK, substitute the new version numbers for your Qt SDK version number.

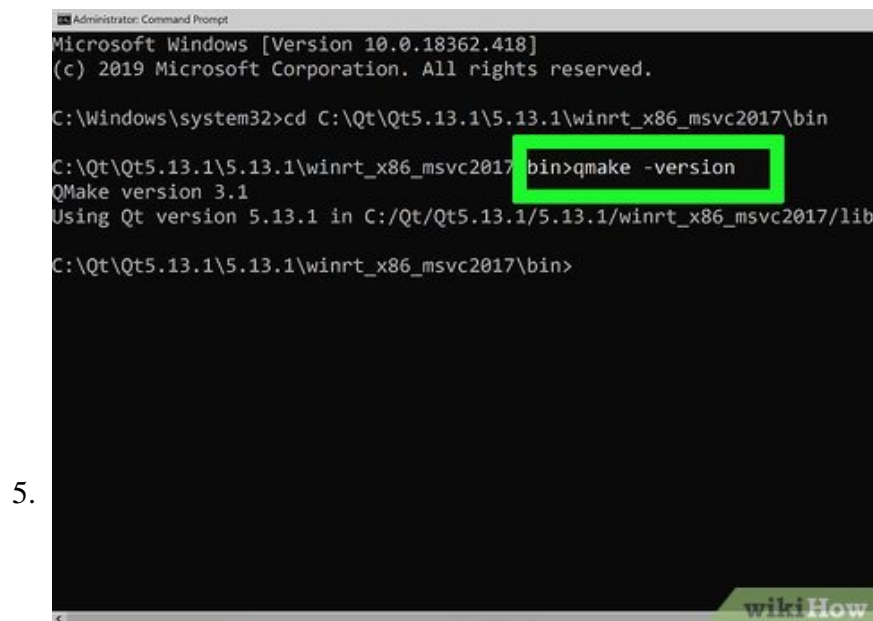


Important: You want to use the version of MinGW included with the Qt SDK in order to build your Qt applications. If another version of the MinGW compiler is installed on the Windows system PATH such as ;C://MinGW/bin you will have to remove it and add the Qt version of MinGW which is included in the Qt SDK. This is critical if you have another version of the MinGW C/C++ compiler installed on your Windows system you must remove the other MinGW from the Windows system PATH, because this will

cause conflicts with the Qt version of MinGW and your Qt applications will not compile and build correctly from the command line properly. Essentially, if you use another version of the MinGW C/C++ compiler your Qt applications you create from the command line will crash and exit with multiple system error messages. You must use the version of MinGW C/C++ compiler included with the Qt SDK.

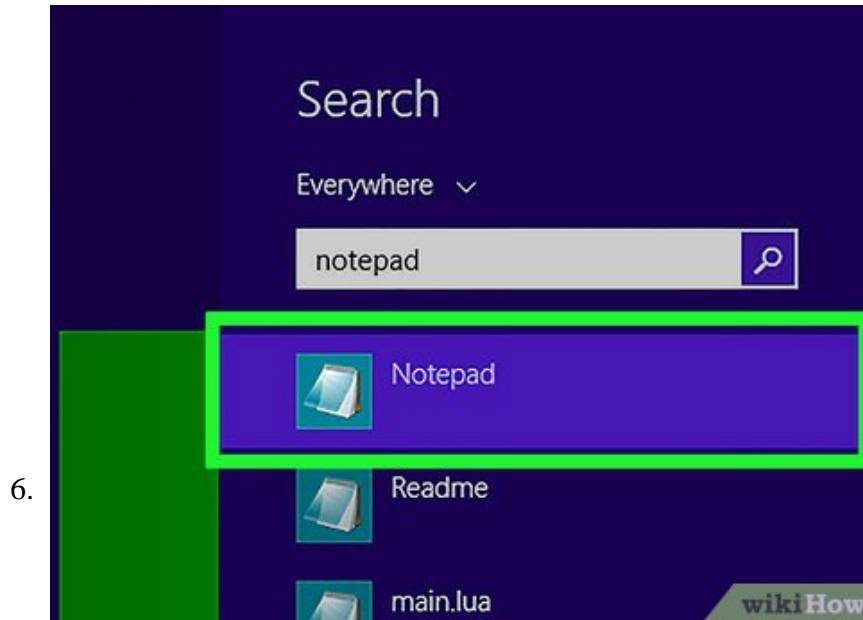


Once the Qt SDK system PATH has been added to the Windows operating system. To check if you have the ability to compile from the Windows command line, open up a command line prompt and issue the following command below.

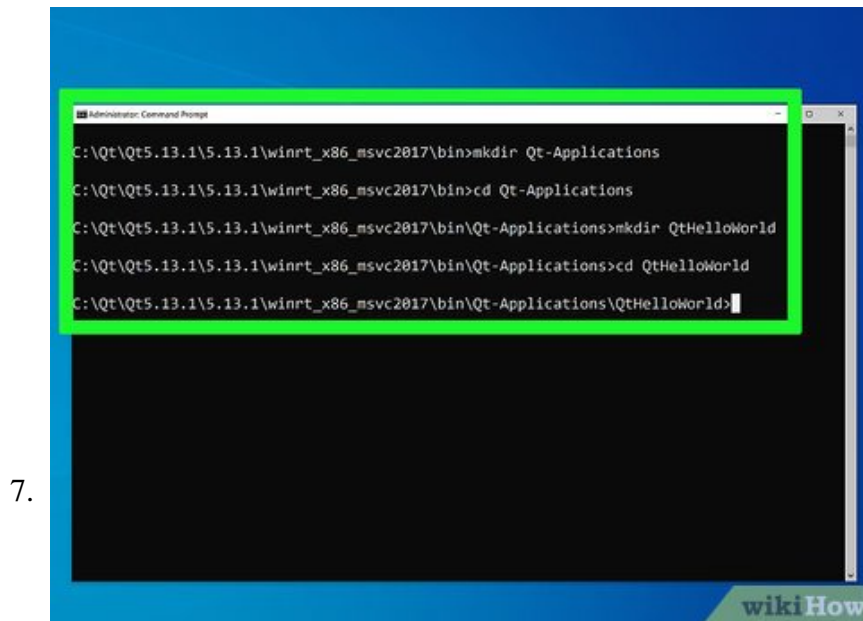


Type/Copy/Paste: qmake -version

1. You should receive a response similar to this:
2. *'QMake version 2.01a*
3. *'Using Qt version 4.8.1 in C:QtSDKDesktopQt4.8.1mingwlib*



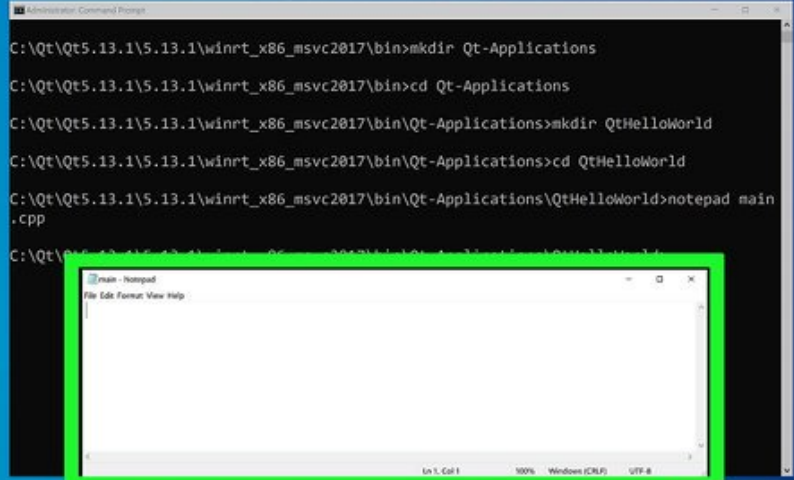
Develop your applications from the command line by using a text editor such as Notepad or WordPad to create and edit the source code and compile your Qt applications from the command line.



Once the Qt SDK is added to the system PATH using the instructions above you can now compile your applications from the command line using the following method by running the following commands. Open up a command line prompt and enter the following commands below.

1. **Type/Copy/Paste:** mkdir Qt-Applications
2. **Type/Copy/Paste:** cd Qt-Applications
3. **Type/Copy/Paste:** mkdir QtHelloWorld
4. **Type/Copy/Paste:** cd QtHelloWorld

8.

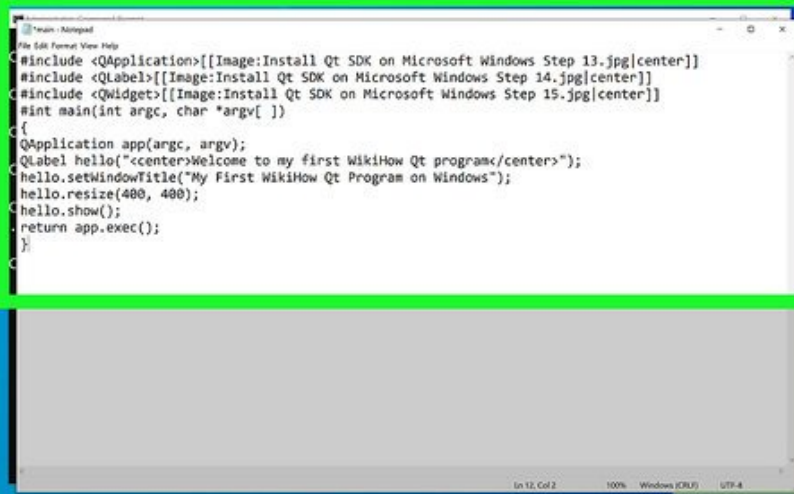


```
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin>mkdir Qt-Applications
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin>cd Qt-Applications
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications>mkdir QtHelloWorld
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications>cd QtHelloWorld
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications\QtHelloWorld>notepad main.cpp
```

While in the QtHelloWorld directory, use a text editor such as Notepad or Wordpad to create your Qt source code type the following

1. **Type/Copy/Paste:** notepad main.cpp
2. make sure you save the Qt source code file as main.cpp
3. **or**
4. **Type/Copy/Paste:** start wordpad
5. using Wordpad as a text editor you will manually have to save the Qt source code file as main.cpp

9.



```
#include <QApplication>[[Image:Install Qt SDK on Microsoft Windows Step 13.jpg|center]]
#include <QLabel>[[Image:Install Qt SDK on Microsoft Windows Step 14.jpg|center]]
#include <QWidget>[[Image:Install Qt SDK on Microsoft Windows Step 15.jpg|center]]
int main(int argc, char *argv[ ])
{
    QApplication app(argc, argv);
    QLabel hello("<center>Welcome to my first WikiHow Qt program</center>");
    hello.setWindowTitle("My First WikiHow Qt Program on Windows");
    hello.resize(400, 400);
    hello.show();
    return app.exec();
}
```

Create the application in the notepad or wordpad text editor, enter the following below

1. **Type/Copy/Paste:**

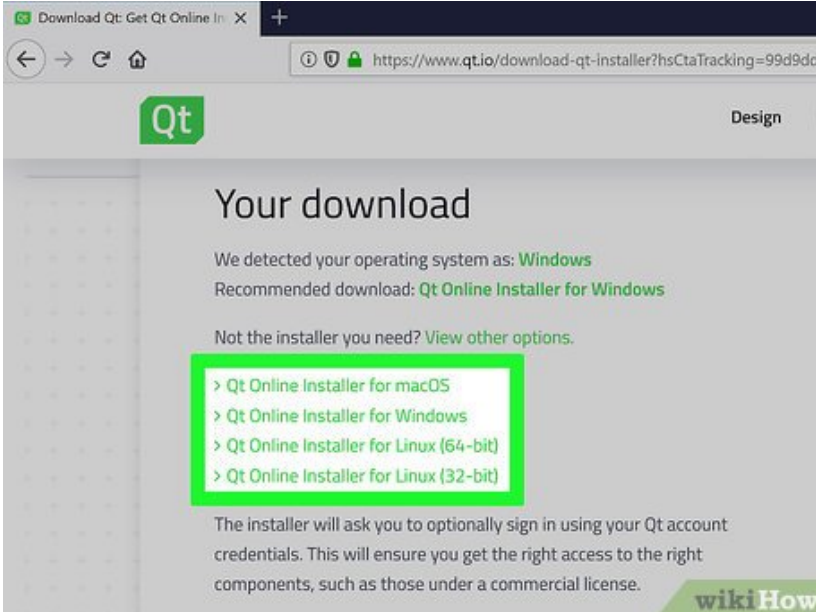
```
#include #include #include #int main(int argc, char *argv[ ]) {
QApplication app(argc, argv); QLabel hello(
"Welcome to my first WikiHow Qt program"); hello.setWindowTitle(
"My First WikiHow Qt Program on Windows"); hello.resize(400, 400); hello.
show(); return app.exec(); }
```

1. Save the source code file as main.cpp
2. While in the QtHelloWorld directory run the following to compile and link the Qt source code
3. **Type/Copy/Paste:** qmake -project
 1. this will create the Qt project file
4. **Type/Copy/Paste:** qmake
 1. this will prepare the Qt project file to be compiled
5. **Type/Copy/Paste:** make
 1. this will compile the Qt source code into an executable program
6. Once the above tasks have been successfully performed and there are no errors The Qt application will be created in the QtHelloWorld debug folder as an executable ending in .exe Change into the debug folder and execute the Qt application by either clicking the application or running the Qt application from the command line.
7. **Type/Copy/Paste:** cd debug
 1. change into the debug folder
8. **Type/Copy/Paste:** QtHelloWorld.exe
 1. and run the newly created executable
9. **Congratulations** you just compiled your Qt application from the command line on Windows.

Method 4 of 6:

Qt 5.0 SDK Installation Instructions:

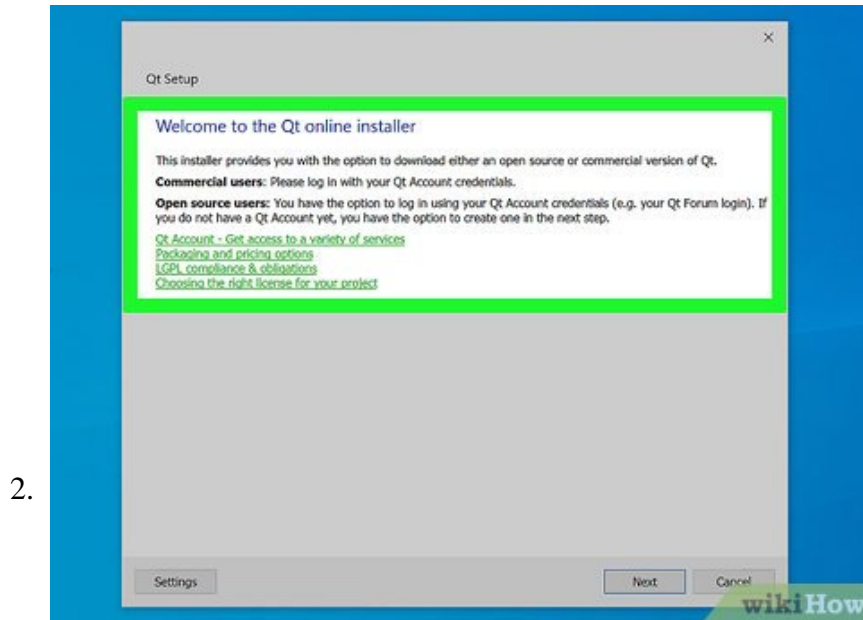
1.



The screenshot shows a web browser window with the URL <https://www.qt.io/download-qt-installer?hsCtaTracking=99d9dd4>. The page features the Qt logo and a 'Design' link. The main heading is 'Your download'. Below it, it states 'We detected your operating system as: Windows' and 'Recommended download: Qt Online Installer for Windows'. A link 'View other options.' is provided. A list of download options is shown, with 'Qt Online Installer for macOS' highlighted by a green box. The other options are 'Qt Online Installer for Windows', 'Qt Online Installer for Linux (64-bit)', and 'Qt Online Installer for Linux (32-bit)'. At the bottom, there is a note: 'The installer will ask you to optionally sign in using your Qt account credentials. This will ensure you get the right access to the right components, such as those under a commercial license.' and a 'wikiHow' logo.

**In order to setup the development environment for the Qt SDK we will have to obtain the Qt SDK.
Preparing your Windows operating system for Qt application development**

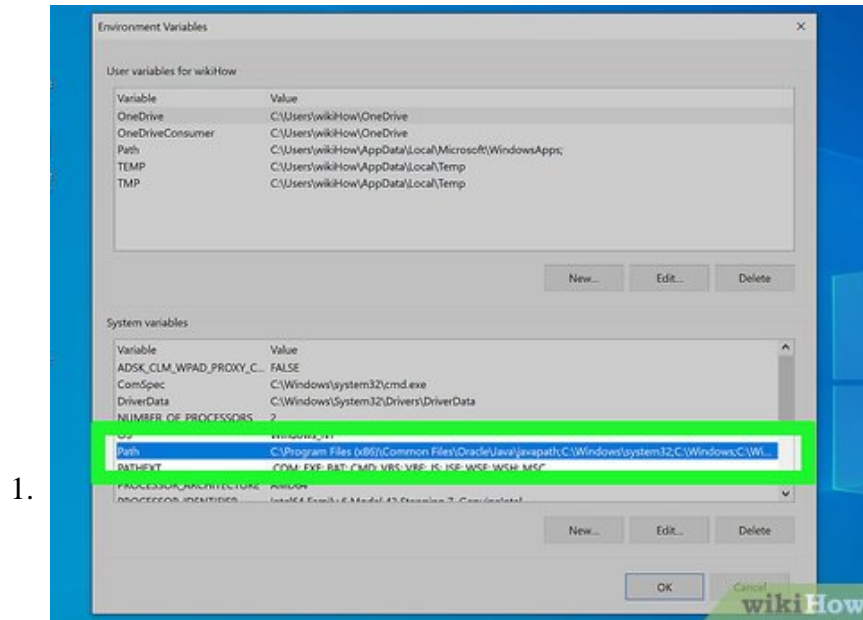
1. Download the Qt SDK. Select the Windows version and be prepared for a long download depending on your Internet speed. I would suggest you select the off-line installation of the Qt SDK if you do not have a very fast internet connection. The complete Qt SDK for the Windows operating system is 1.7 GB which can take over 6 hours on a slow connection.



Install the Qt SDK by clicking on the Qt SDK executable. Once the Qt SDK executable is installed you will need to edit the Windows system PATH so that the Windows operating system can find the Qt commands from the command line. Be careful editing your Windows system PATH.

Method 5 of 6:

Edit the Windows Vista/Windows 7 PATH:

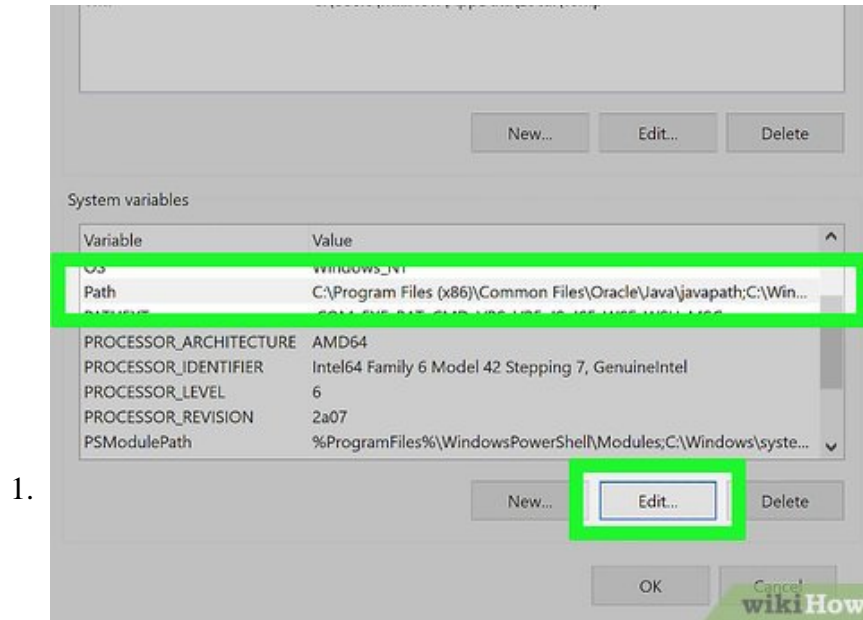


Follow the steps below.

1. Click on the Start Button
2. Click on Control Panel
3. Click on System and Maintenance
4. Click on System
5. Click on Advanced System Setting
6. Click on Environmental Variables
7. Click on PATH to edit system wide variables
8. Click OK when finished

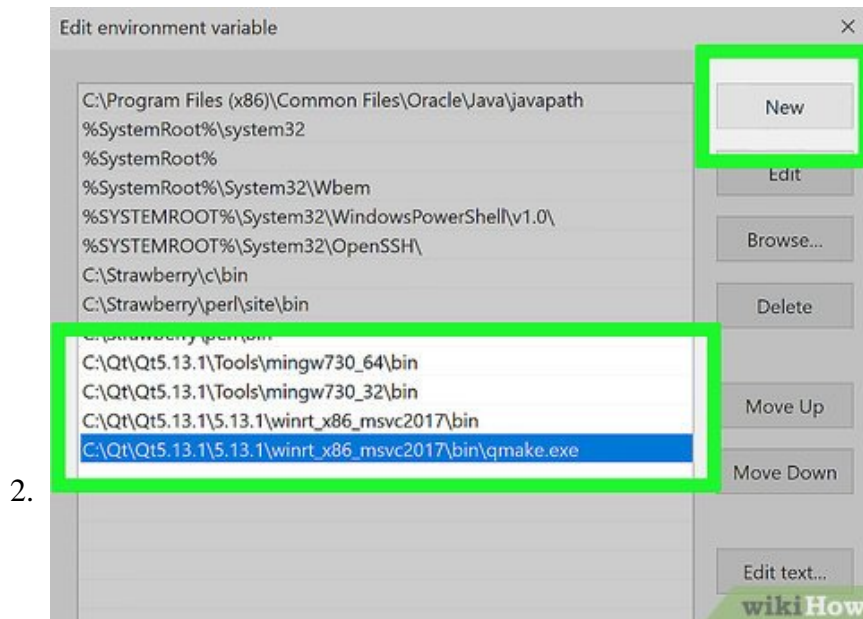
Method 6 of 6:

Edit the Windows 8 PATH:



Follow the steps below.

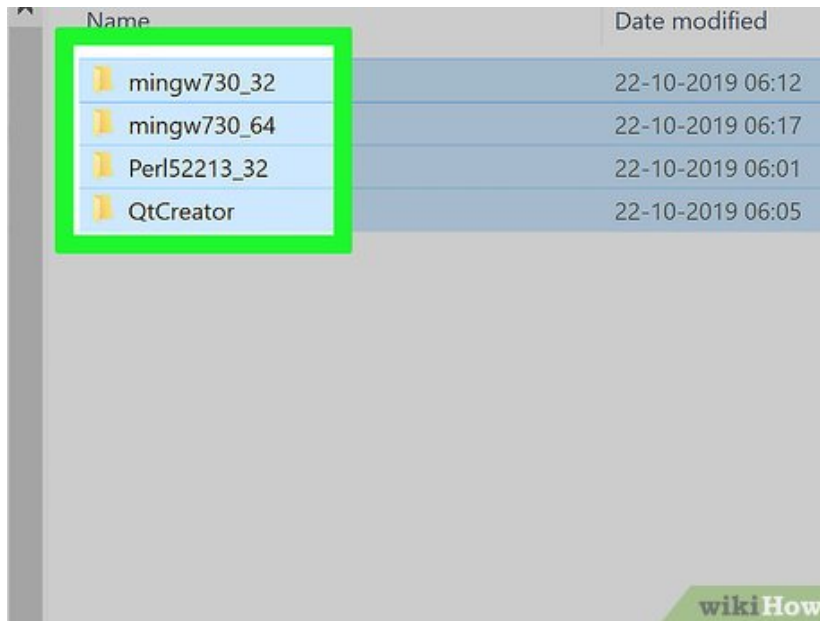
1. Click on the Folder icon located on the bottom task bar next to the Internet Explorer icon
2. Scroll down to Computer
3. Right click with your mouse on Properties
4. Click on Advanced System Setting
5. Click on Environmental Variables
6. Click on PATH to edit system wide variables
7. Click OK when finished



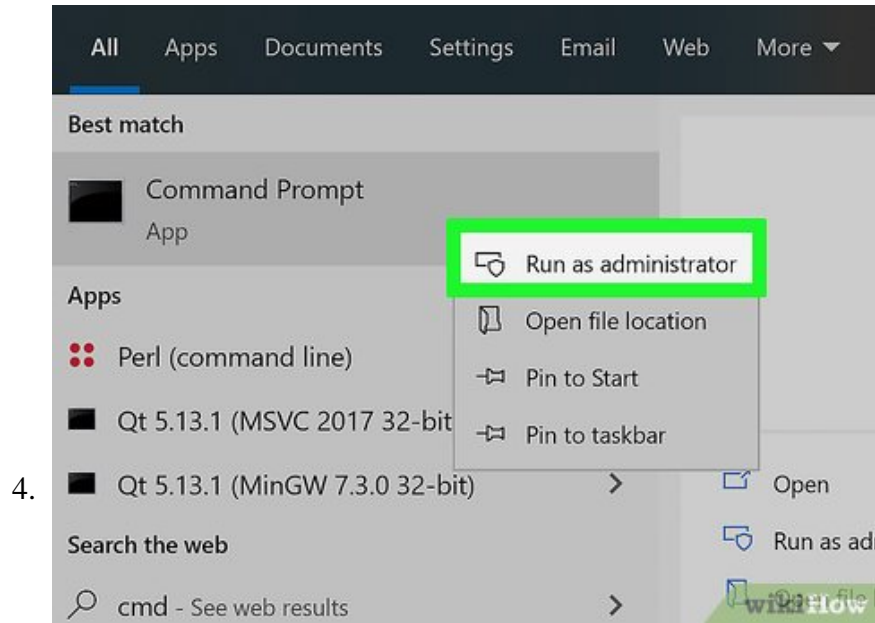
Add the following to the Windows 8 System PATH

1. Type/Copy/Paste:;C:\Qt\Qt5.0.25.0.2mingw47_32bin;C:\Qt\Qt5.0.2ToolsMinGWbin;

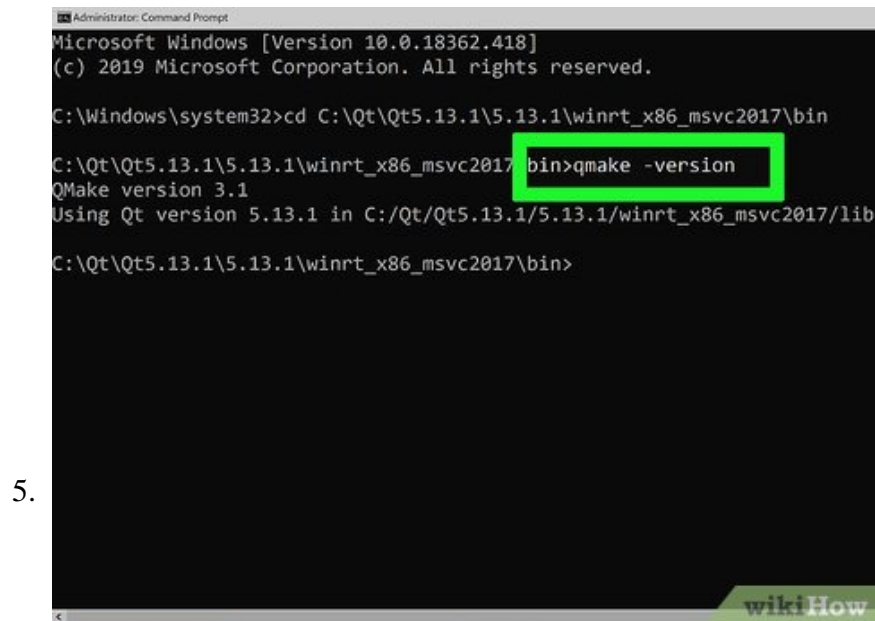
2. this will set and notify the Windows operating system to compile Qt SDK applications from the command line. The numbers **5.0.2** denote the version number of the Qt SDK which changes with each new upgrade of the Qt SDK, substitute the new version numbers for your Qt SDK version number.



Important: You want to use the version of MinGW included with the Qt SDK in order to build your Qt applications. If another version of the MinGW compiler is installed on the Windows system PATH such as ;C://MinGW/bin you will have to remove it and add the Qt version of MinGW which is included in the Qt SDK. This is critical if you have another version of the MinGW C/C++ compiler installed on your Windows system you must remove the other MinGW from the Windows system PATH, because this will cause conflicts with the Qt version of MinGW and your Qt applications will not compile and build correctly from the command line properly. Essentially, if you use another version of the MinGW C/C++ compiler your Qt applications you create from the command line will crash and exit with multiple system error messages. You must use the version of MinGW C/C++ compiler included with the Qt SDK.

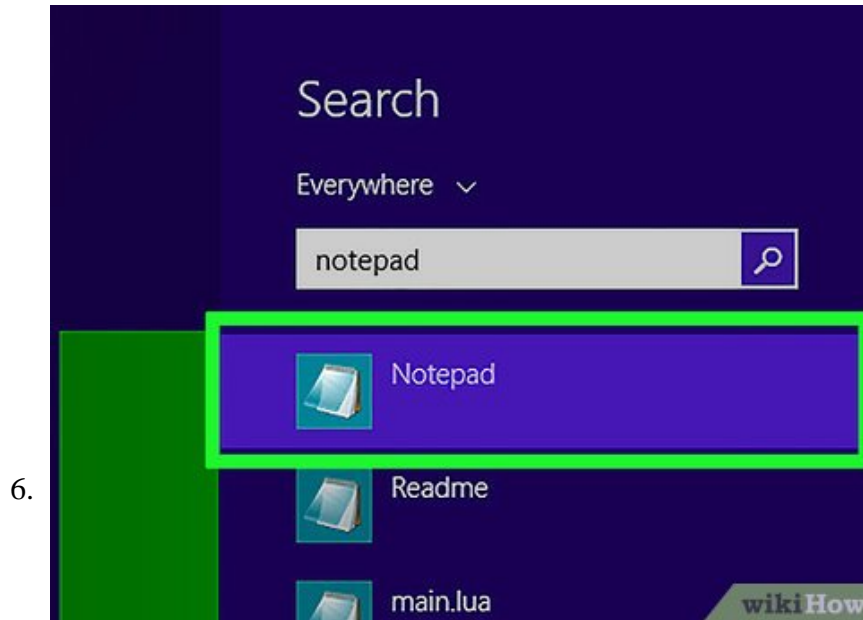


Once the Qt SDK system PATH has been added to the Windows operating system. To check if you have the ability to compile from the Windows command line, open up a command line prompt and issue the following command below.

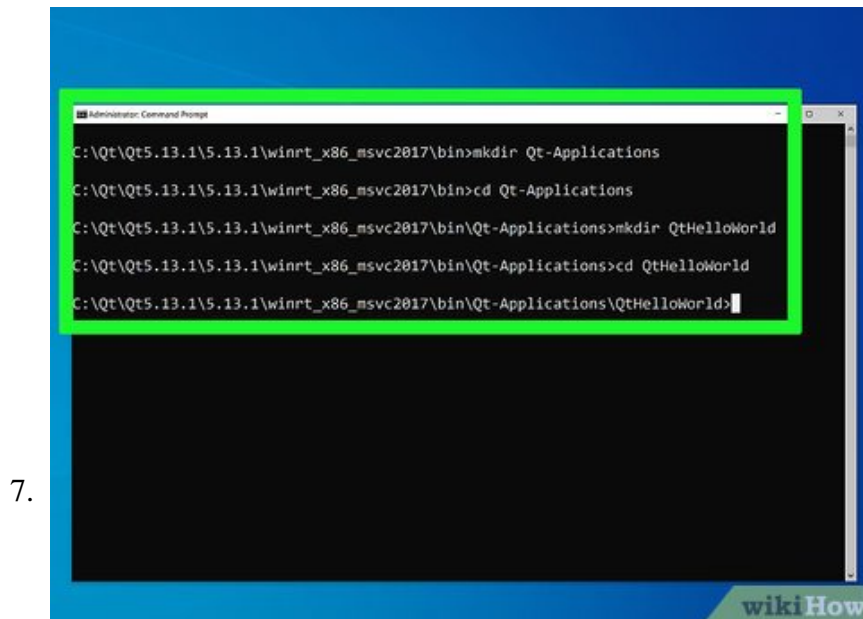


Type/Copy/Paste: qmake -version

1. You should receive a response similar to this:
2. *'QMake version 3.0*
3. *'Using Qt version 5.0.2 in C:QtQt5.0.2mingwlib*



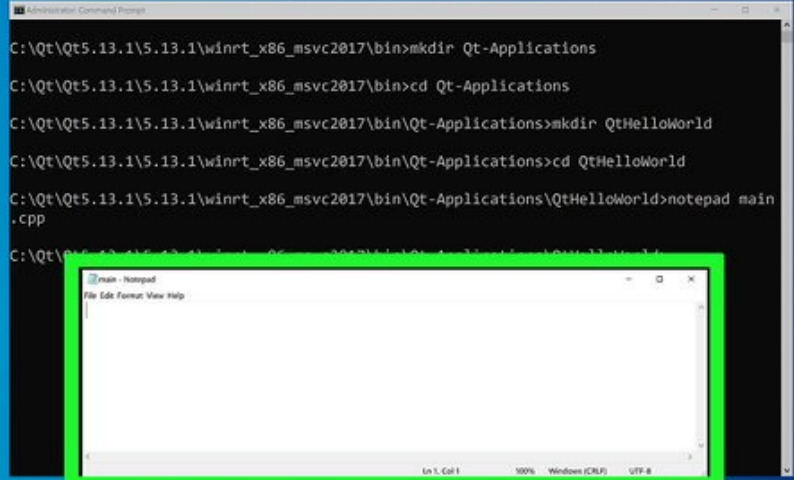
Develop your applications from the command line by using a text editor such as Notepad or WordPad to create and edit the source code and compile your Qt applications from the command line.



Once the Qt SDK is added to the system PATH using the instructions above you can now compile your applications from the command line using the following method by running the following commands. Open up a command line prompt and enter the following commands below.

1. **Type/Copy/Paste:** mkdir Qt-Applications
2. **Type/Copy/Paste:** cd Qt-Applications
3. **Type/Copy/Paste:** mkdir QtHelloWorld
4. **Type/Copy/Paste:** cd QtHelloWorld

8.

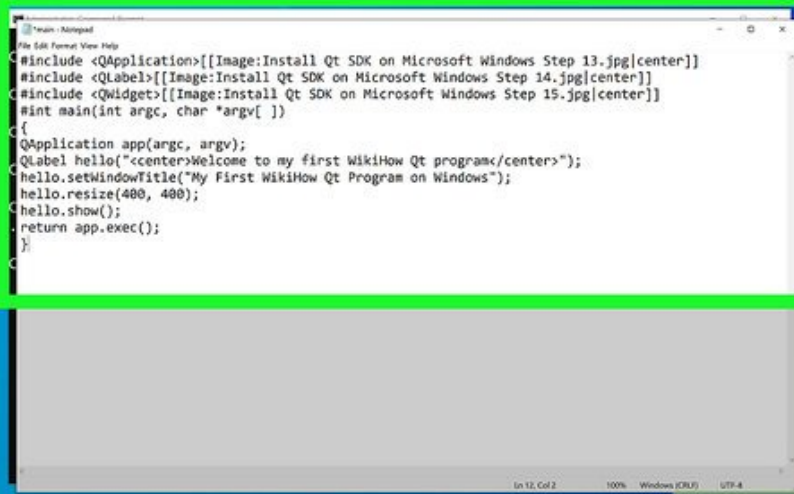


```
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin>mkdir Qt-Applications
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin>cd Qt-Applications
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications>mkdir QtHelloWorld
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications>cd QtHelloWorld
C:\Qt\Qt5.13.1\5.13.1\winrt_x86_msvc2017\bin\Qt-Applications\QtHelloWorld>notepad main.cpp
```

While in the QtHelloWorld directory, use a text editor such as Notepad or Wordpad to create your Qt source code type the following

1. **Type/Copy/Paste:** notepad main.cpp
2. make sure you save the Qt source code file as main.cpp
3. **or**
4. **Type/Copy/Paste:** start wordpad
5. using Wordpad as a text editor you will manually have to save the Qt source code file as main.cpp

9.



```
#include <QApplication>[[Image:Install Qt SDK on Microsoft Windows Step 13.jpg|center]]
#include <QLabel>[[Image:Install Qt SDK on Microsoft Windows Step 14.jpg|center]]
#include <QWidget>[[Image:Install Qt SDK on Microsoft Windows Step 15.jpg|center]]
int main(int argc, char *argv[ ])
{
    QApplication app(argc, argv);
    QLabel hello("<center>Welcome to my first WikiHow Qt program</center>");
    hello.setWindowTitle("My First WikiHow Qt Program on Windows");
    hello.resize(400, 400);
    hello.show();
    return app.exec();
}
```

Create the application in the notepad or wordpad text editor, enter the following below

1. **Type/Copy/Paste:**

```
#include #include #include #int main(int argc, char *argv[ ]) {
QApplication app(argc, argv); QLabel hello(
"Welcome to my first WikiHow Qt program"); hello.setWindowTitle(
"My First WikiHow Qt Program on Windows"); hello.resize(400, 400); hello.
show(); return app.exec(); }
```

1. Save the source code file as main.cpp
2. While in the QtHelloWorld directory run the following to compile and link the Qt source code
3. **Type/Copy/Paste:** qmake -project
 1. this will create the Qt project file
4. However, in Qt 5.0 SDK you must use a text editor and add the following to your generated *.pro file, use a text editor such as nano or gedit and issue the following commands:
5. **Type/Copy/Paste:** notepad QtHelloWorld.pro
6. Your generated QtHelloWorld.pro should look similar to this:

```
TEMPLATE = app TARGET = QtHelloWorld #INCLUDEPATH += . # Input SOURCES +=
main.cpp
```

1. Edit your generated QtHelloWorld.pro file to look like this:

```
TEMPLATE = app TARGET = QtHelloWorld QT += core gui QT += widgets
#includepath += . # Input SOURCES += main.cpp
```

1. Once you add the following lines to the QtHelloWorld.pro file under the TARGET keyword, then run qmake
2. **Type/Copy/Paste:** QT += core gui
3. **Type/Copy/Paste:** QT += widgets
 1. Save the file and exit
4. **Type/Copy/Paste:** qmake
 1. This will create the Qt make file
5. **Type/Copy/Paste:** make
 1. This will compile the Qt make file on your system into an executable program. At this point, providing that there are no errors the file should compile into an executable program.
6. Once the above tasks have been successfully performed and there are no errors The Qt application will be created in the QtHelloWorld debug folder as an executable ending in .exe Change into the debug folder and execute the Qt application by either clicking the application or running the Qt application from the command line.
7. **Type/Copy/Paste:** cd release
 1. change into the release folder
8. **Type/Copy/Paste:** QtHelloWorld.exe
 1. and run the newly created executable
9. **Congratulations** you just compiled your Qt application from the command line on Windows.

You finished reading the article "**How to Install Qt SDK on Microsoft Windows**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

